BigData Full Stack Training Plan

Chinnasamy

Haaris Infotech

Purpose

The purpose of this document is the complete training content on BigData stack by Haaris Infotech.

Prerequisite

Candidate attending the training should have a basic knowledge on any programming language

Duration

Total Hours: 60 Hours
Four hours each day

Copyright

No	Name	Company	Date	Comments
1	Chinnasamy, Shoaib	Haaris Infotech	October 21, 2020	Version 1.0

Project: A live project of how each of the API's are used in the industry.

Use cases covered:

[1] A csv file format of three hundred columns will be used as a dataset.
[2] Consuming and operating two csv files (each of 3 MB) that are produced every second through spark streaming.
[3] Ten to fifteen transformation on a single job. Efficiently optimize and fine tune on all the transformations.
[4] Architectural sharing of data between spark jobs.

Hands-on/Lecture Ratio:
The course is 60 % hands-on, 40 % discussion, with the longest discussion segments lasting 20 minutes.

Note to participants:

[*] All content in this course will be a hands-on session.
[*] All slides of the course will be given to candidates.
[*] Source code of all examples tried out in the session will be provided.

Training Developers Environment:

[*] The training programs would be given as a intellij project.
So I would need in a internet connection for the maven execution. The maven would download a lot of jars from the internet.
[*] Download sbt and install it.
http://www.scala-sbt.org/download.html
[*] Download IntelliJ and install it. https://www.jetbrains.com/idea/
[*] Eclipse Mars, JDK 8, Spark 1.6.0 or Spark 2.0 installation on their respective OS.
[*] Any linux or unix flavor box is needed for the trainees to do the cluster setup of spark.
The box should have an internet connection, JDK 8 and Spark 2.0 or
Spark 1.6 installed in it.
[*] If the box is not available, then the VM of Ubuntu Linux is needed. In order to run Linux VM, Oracle Virtualbox or VMWare is needed.
[*] JDK_HOME and PATH variable to the JDK 1.8 should be set.
[*] The trainer has a MAC laptop, so infrastructure should be provided to
connect MAC laptop to the screen.

Day 1

Big Data Conceptuals

- ❏ What is Big Data?
- ❏ The need for Big Data
- ❏ Why Big Data now?
- ❏ Myths of Big Data
- ❏ Tabular representation of data unit measurement.
- ❏ Datawarehouse and its need
- ❏ Datamarts, OLTP and OLAP systems
- ❏ ETL needs in the big data space
- ❏ The need to move from datawarehouse to big data

Day 2

- ❏ Is one petabyte big data ?
- ❏ Types of Architectures in Big Data
  - ❏ Lambda Architecture
  - ❏ Kappa Architecture
  - ❏ Zeta Architecture
  - ❏ Seda Architecture
  - ❏ NoSQL Store and high throughput messaging system

Day 3

- ❏ Illustration about CAP theorem
- ❏ NoSQL, Types of NoSQL.
- ❏ Problems with large-scale systems
- ❏ Installation of all softwares, the developer environment etc.., will be done in parallel

Day 4

- ❏ HDFS and HADOOP

    - ❏ Introduction
    - ❏ Motivation for Hadoop
    - ❏ Introduction to Hadoop Processes/Architecture
    - ❏ Hands on: Setting Up Hadoop Cluster
    - ❏ Hadoop: Basic concepts
    - ❏ HDFS
    - ❏ Why HDFS ?
    - ❏ AWS S3 vs HDFS

Day 5

- ❏ HDFS Architecture
- ❏ Using HDFS and hdfs commands
- ❏ Hands-on Exercise: Using HDFS
- ❏ Concepts behind MapReduce
- ❏ How Hadoop cluster operates
- ❏ Hands-on Exercise: Running a MapReduce job
- ❏ Writing a MapReduce program
- ❏ Examining the MapReduce program
- ❏ Hands-on Exercise: Write a MapReduce program
- ❏ The New MapReduce API
- ❏ Mapper
- ❏ Reducer
- ❏ Comparing spark with Map-Reduce code

Day 6

HIVE

- ❏ Introduction to Hive
- ❏ Hive Basics
- ❏ Write the auth data in the Hive tables. When to you use internal tables and external tables.
- ❏ Then load, index and drop the tables
- ❏ Writing different queries on the top of the data.
- ❏ Playing with hue

SQOOP

- ❑ The need for SQOOP
- ❑ Importing and exporting data with SQOOP
- ❑ Use of KafkaConnect rather than using Flume
- ❑ How SQOOP differs from KafkaConnect

Day 7

Scala

[1] Introduction to Scala

[1.1] Why Scala?

[1.2] What is Scala?, Introducing Scala, Installing Scala, Journey - Java to Scala

[1.3] First Dive - Interactive Scala, Writing and Compiling Scala Programs

[1.4] Scala - REPL

[1.5] Scala Basics and Scala Basic Types

[1.6] Defining functions - Functions are first class citizens

[1.7] Imperative languages vs Functional Languages

[1.8] IDE for Scala, Scala Community.

[1.9] About the IntelliJ IDE. Setting up the IDE for the scala development.

[2] Scala Essentials

[2.1] Immutability in Scala.

[2.2] Semicolons and return statement.

[2.3] Method Declaration, Literals, Reserved Words, Operators, Precedence Rules, If
statements, While Loops, Do-While Loops, Conditional Operators.

[2.4] Enumerations.

[2.5] Factory Pattern using match keyword

Day 8

Usage of Object orientation in Scala

[3] Traits and OOPs in Scala

[3.1] Traits - Traits as Mixins, Stackable Traits.

[3.2] Creating Traits Basic OOPS - Class and Object Basics.

[3.3] Class Constructors, Nested Classes, Visibility Rules.

[4] Functional Programming in Scala

[4.1] Topics - What is Functional Programming?, Functional Literals and Closures,

[4.2] Recursion, Tail Calls,

[4.3] Functional Data Structures,

[4.4] Implicit Function Parameters - Implicit values, Implicit Conversions and Implicit

classes.

[4.5] Call by Name, Call by Value.

[5] Functional Programming

[5.1] Map Transformation.

[5.2] Writing a functional literal (lambda expression) in a map transformation.

[5.3] map, flatMap, reduce, filter, head, take, drop, reduceLeft, fold, foldLeft, zip transformations.

[5.4] Writing different types of functions.

[6] Variable Arguments

[6.1] Discussion on the _* type.

[6.2] Usage of underscore in different places.

Day 9

[7] Collections

[7.1] List

[7.2] Set

[7.3] Tuple

[7.4] Range

[7.5] Arrays

[7.6] Mutable

[7.7] Immutable

[7.8] Parallelized Collections

[7.9] Collection Transformations

[8] Currying Functions

[8.1] Detailed study and usage of currying and partial applied functions.

[9] Build

[9.1] Elucidation on maven and sbt

[9.2] Coding a maven pom file

[9.3] Coding a sbt

[10] Leftovers

[10.1] Bounded Types

[10.2] isInstanceOf and asInstanceOf

[10.3] Usage of annotations - concise code

[10.4] Sealed classes

[10.5] Option Class
[10.6] Building a jar using maven or sbt

Day 10

Spark (The Spark version covered is the latest version of Spark - 2.0)

JDK 8 - Quick Introduction

- Functional Programming with Java
- Lambda expressions and Functional Interfaces in Java

Scala - Introduction

- Objects and Classes
- val, var, functions, currying, implicits
- traits, actors and file manipulations

Core Spark

- Introduction to Apache Spark
- What is Spark ? Explain about the modules in spark
- Spark-Shell - scala and python REPL
- Spark Internals - The Driver program, master, workers, executors and the tasks
- SparkSession- The Umbrella API for all context
- Running spark in a standalone mode
- Spark UI and monitoring a job
- Functional programming with Spark
- Map-reduce and Spark advantages over Map-reduce.

RDD

- What is an RDD ?
- Laziness in RDD Evaluation
- Different ways of creating an RDD
- Types of RDD's - PairRDD, DoubleRDD

    - RDD Operations
    - Partitions - The core of RDD
    - textFiles, wholeFiles

Day 11

Running Spark on a Cluster

- Overview
- A Spark Standalone Cluster
- The Spark Standalone Web UI
- Installing and configuring a cluster

Operations in Spark

- Spark Configuration and the Spark Context
- Configuring spark properties
- RDD Operations - Transformation and Actions
    - map, flatMap, repartition, coalesce, glom, reduce, cartesian, pipe, sample, distinct, mapPartitions, mapPartitionsWithIndex
    - Map, filter, distinct, collect, take operations
- Joining two RDD's
- Storage levels supported in spark
- Programming with a partition and use of custom partitioners
- Accumulators and Broadcast variables
- Checkpointing an RDD
- Spark deployment plans
- Spark History Server

Reading Data from External Sources

- JdbcRdd - Read data from mysql
- Connecting and reading data from mongodb

Day 12

Caching and Persistence

- RDD Lineage
- Caching Overview
- Distributed Persistence

SparkSQL

- The DataFrame Abstraction
- Elucidate on SparkSQL
- Dataframe manipulation on top of json
- The temp table abstraction on top of DataFrame Schema
- SQL manipulation on top of parquest files
- Dataframes caching
- Connecting dataframes to relational database

Spark Streaming

- Kafka and the need
- Basic read from a socket
- Spark Streaming from kafka
- Windowing operation in streaming
- Developing streaming applications
- Writing a custom receiver
- Spark Structured Streaming

Day 13

Advanced Topics

- Spark SQL with Hive
- The new Dataset API
- Connecting Spark with HBASE
- Working with nested data
- Spark with Alluxio
- Custom Accumulators
- Writing custom RDD
- Writing custom partitioner
- Internals of persistence API. How spark manages persistence internally. (Drilling down the source code)
- Connecting spark with cassandra and ingesting data into cassandra

Spark Performance Tuning

Various strategies to adopt to performance tune your spark application.
Introduction to various variables in Spark like shared variables.
Broadcast variables and learning about accumulators.
Common performance issues and troubleshooting the performance problems.

Maven would be used as the build tool to download the dependencies. IntelliJ would be the IDE to develop the applications and examples.

Day 14

Zookeeper and Kafka - Streaming

Zookeeper Overview
- Why and What is Distributed Service and why we need Zookeeper
- CAP - Brewer's Theorem
- Systems that use zookeeper as the underlying storage

Installing Zookeeper
- System requirements and installing and managing a Zookeeper cluster

Zookeeper Architecture
- Quorum
- Epoch
- Znode
- Session
- Watcher
- Persistent Znode
- Ephemeral Znode
- Sequential Znode

Hands on with Zookeeper CLI (Call level Interface)
- Create
- Get
- Set
- Delete

Programmatically accessing zookeeper using Java

Java code to create,get,set on a znode

Zookeeper Administration

Configuring Zookeeper
Managing Zookeeper Storage
Remotely Connecting to Zookeeper
Logging

How Zookeeper Works

Leader Election
Locks
Queues

Kafka with Zookeeper

Installing Kafka and running Kafka on top of zookeeper
How kafka interacts internally with Zookeeper

Requirements of Kafka

Real time analytics
Data ingestion
Case studies

Kafka architecture

Core concepts
Kafka Design
Log Compaction
Message compaction
Replication
Message flow
High Availability and Consistency
Resource Management
Topics
Partitions
Replicas
Producers
Consumers
Brokers
Segment

- Offset
- Leader
- Follower

Day 15

Kafka Internals

- Last Commit Offset
- In-Sync Replicas
- High Watermark
- Log end offset
- Single and Multiple Consumer with Multiple Consumer Group
- Consumer rebalancing
- Group Coordinator and Group Leader Strategy

Kafka Development- Java/Scala Coding

- Architecture
- Hardware specs
- Deploying Deep Dive into Kafka Cluster
- Understanding the components of Kafka cluster
- Installation of Kafka Cluster
- Configuring Kafka Cluster
- Producer of Kafka
- Consumer of Kafka
- Producer and Consumer in Action
- Hands on code with Java and Scala
- Replication and Compression
- Subscribing to topics
- Assignment to topic partitions

Kafka Cluster

- Install Kafka
- Set up a Kafka -
  - A single node- A single broker cluster
  - A single node - Multiple broker clusters
  - Multiple nodes - Multiple broker clusters

- Kafka Operations and Performance Tuning

Kafka Streams

- KStreams
- KTable
- All transformations with KStreams and KTable-
  - map(),mapValues(),filter(),flatMap(),groupBy(),groupByKey(),foreach(), peek(),writeAsText(),print()
- Windowing

Kafka Connect

- Data transfer through connect to HDFS and kafka topics.
- Working with Kafka Logs
- Operationalizing Kafka Securing Kafka
- Security Overview
- Configuring Kafka Security

Encryption and Authentication using SSL

- Authentication using SASL
- Authorization and ACLs
- Incorporating Security Features in a Running Cluster
- ZooKeeper for HA Hands On
- Using Kafka Connect to move data
- Monitoring and Alerting using Kafka Tools
- Set up authentication for Kafka
- Authentication via SSL & Kerberos through SASL
- Authorization, permissions and ACLs setup
- Set up Encryption